

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-244020

(43)公開日 平成5年(1993)9月21日

(51)Int.Cl. ¹	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 M 13/22		7259-5 J		
G 0 6 F 11/10	3 3 0 F	7313-5 B		
H 0 4 L 1/00		B 6942-5 K		

審査請求 有 請求項の数12(全 14 頁)

(21)出願番号 特願平4-270145

(22)出願日 平成4年(1992)10月8日

(31)優先権主張番号 7 9 2 1 1 3

(32)優先日 1991年11月14日

(33)優先権主張国 米国(US)

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーションINTERNATIONAL BUSIN
ESS MACHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72)発明者 ミゲール・マリオ・ブラウム

アメリカ合衆国95119、カリフォルニア州

サン・ノゼ、マニラ・ドライブ 341番

地

(74)代理人 弁理士 頓宮 孝一 (外3名)

最終頁に続く

(54)【発明の名称】 インターリーブ式の誤り訂正符号を復号化するための復号器能力強化方法

(57)【要約】 (修正有)

【目的】 インターリーブ式の誤り訂正符号を復号化する復号器の、復号化能力を強化する。

【構成】 nバイト長の複数のコード化済み符号語を行方向に配列して深さnの複数の矩形ブロックを形成し

(nとnは互いに素)、ブロック毎順番にインターリーブして伝送する。インターリーブはブロックの斜め方向に実行される。受信側では、受信データをデインターリーブし復号するが、誤りバイト数が4以上でECC能力を超えている行1、4の符号語にフラグ付けをし、訂正可能である行0の誤りバイト2、3に斜め方向に隣接する行1のバイト3、4を消滅痕として行1を訂正可能にし訂正する。また、行3の誤りバイト0、10に斜め方向に隣接する行4のバイト1、11を消滅痕として、同様に誤り訂正を実行する。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0			b ₁	b ₂													
F 1				b ₁	b ₂				b ₃				R				
2			R							b ₃						b ₁	
3	b ₂										b ₃						b ₁
F 4	b ₁	b ₂							R			b ₃					
5		b ₁	b ₂										b ₃				

【特許請求の範囲】

【請求項1】 誤り訂正符号（ECC符号）に従いインターリーブされた複数のコード化済シンボルから成る複数の符号語を復号化する復号器の能力を、冗長度を増大することなく強化する、復号器能力強化方法において、全ての符号語を復号化し、全ての訂正可能な符号語の中の全ての訂正可能な誤りを訂正し、前記ECC符号の誤り訂正能力を超える符号語のみをフラグで標識付けしてフラグ付符号語とする、復号化ステップと、

前記フラグ付符号語の各々について、当該フラグ付符号語に対して時間的に隣接した少なくとも1つの符号語を検査して、その隣接符号語中の誤り状態にあるロケーションであって、訂正がなされてその復号化が成功したロケーションを識別する、検査ステップと、

当該フラグ付符号語中のロケーションのうち、その隣接符号語中の誤り状態にあったロケーションに対して斜め方向に時間的に隣接した特定ロケーションに存在する誤りを暫定的に消滅痕であると宣言する、宣言ステップと、

前記復号化ステップを再実行して、当該フラグ付符号語を訂正して復号化することを試みる、再実行ステップと、

を含んでいることを特徴とする復号器能力強化方法。

【請求項2】 請求項1記載の方法において、該方法は更に、

n の値と λ の値とを、それら値が互いに素であるという条件を満足する任意の所望の整数値に選定するステップと、

複数の符号語が矩形ブロックの中に並ぶように符号化を行ない、その際に各ブロックにおいて、個々の符号語がそのブロックの中の各々の横列を成すようにすると共に、夫々の横列を成す符号語の間で互いに対応するシンボルどうしが縦方向に位置が揃うようにし、更に、各ブロックが、シンボル n 個分の長さの符号語が入行並んだものから成るようにする、符号化ステップと、

各ブロック中の連続する複数の符号語の複数のシンボルを、1つの連続した斜めのシーケンスに沿ってインターリーブすることにより、位相幾何学的に円環体を成すアレイを形成し、それによって、全てのバースト誤りが、前記アレイ中のそれらバースト誤りの発生位置にかかわらず符号語から符号語へと連続するようにする、インターリーブステップと、

を含んでいることを特徴とする方法。

【請求項3】 請求項1記載の方法において、該方法は更に、

複数の符号語が矩形ブロックの中に並ぶように符号化を行ない、その際に各ブロックが、シンボル n 個分の長さの横列が縦に λ 本並んだものから成るようにすると共に、各ブロックにおいて、夫々の横列の間で互いに対応するシンボルどうしが縦方向に位置が揃うようにする、

符号化ステップと、

$a(i, j)$ を各ブロックの $i+1$ 行、 $j+1$ 列（ただし、 $0 \leq i \leq \lambda-1$ 、 $0 \leq j \leq n-1$ ）のシンボルとして、各ブロックの中の複数のシンボルを、

・ $a(0, 0)$ の書込みを行ない、

・ $a(i, j)$ の書込みを行なったならば、続いて $a((i+1) \lambda, (j+1) n)$ の書込みを行ない、

・ $a(\lambda-1, n-1)$ の書込みを行なったならば、終了する

10 という規則によって規定されるシーケンスに沿って、斜め方向にインターリーブすることにより、位相幾何学的に円環体を成すアレイを形成し、それによって、全てのバースト誤りが、前記アレイ中のそれらバースト誤りの発生位置にかかわらず符号語から符号語へと連続するようにする、インターリーブステップと、

を含んでいることを特徴とする方法。

【請求項4】 請求項1記載の方法において、前記検査ステップで、前記フラグ付符号語に対する先行符号語と後続符号語との誤りの有無を検査し、且つ、前記宣言ステップで、前記先行符号語中の誤り状態ロケーションと前記後続符号語中の誤り状態ロケーションとの両方に対して時間的に斜めに隣接する、該フラグ付符号語中のロケーションを消滅痕であると宣言するようにしたことを特徴とする方法。

【請求項5】 請求項1記載の方法において、該方法は更に、

前記フラグ付符号語が、前記再実行ステップにおいて訂正不可能であったならば、訂正不可能誤りの宣言をする、訂正不可能誤り宣言ステップと、

30 所定のプロトコルを起動するステップと、

を含んでいることを特徴とする方法。

【請求項6】 請求項1記載の方法において、前記検査ステップに続き、所定のプロトコルに従って、誤り状態ロケーションであると識別したロケーションを分離して誤り集合とする、誤り集合形成ステップを含んでおり、且つ、前記宣言ステップでは、異なった夫々の誤り集合中の誤り状態ロケーションに対して隣接した特定のロケーションをシーケンシャルに消滅痕として宣言するようにしたことを特徴とする方法。

40 【請求項7】 ECC符号に従いインターリーブされた複数のコード化済シンボルから成る複数の符号語を復号化する復号器の能力を、冗長度を増大することなく強化する、復号器能力強化方法において、

50 複数の符号語が矩形ブロックの中に並ぶように符号化を行ない、その際に各ブロックにおいて、個々の符号語がそのブロックの中の各々の横列を成すようにすると共に、夫々の横列を成す符号語の間で互いに対応するシンボルどうしが縦方向に位置が揃うようにする、符号化ステップと、

3

の際に、そのブロック中の連続する複数の符号語の複数のシンボルを、1つの連続した斜めのシーケンスに沿ってインターリーブすることにより、位相幾何学的に円環体を成すアレィを形成し、それによって、全てのバースト誤りが、前記アレィ中のそれらバースト誤りの発生位置にかかわらず符号語から符号語へと連続するようにするインターリーブステップと、

を含んでいることを特徴とする復号器能力強化方法。

【請求項8】 請求項7記載の方法において、前記矩形ブロックの各々が、シンボル n 個分の長さの符号語が入行並んだものから成るようにし、且つ、インターリーブステップにおいて、 $a(i, j)$ を各ブロックの $i+1$ 行、 $j+1$ 列（ただし、 $0 \leq i \leq \lambda-1$ 、 $0 \leq j \leq n-1$ ）のシンボルとして、各ブロック中の複数のシンボルを、

- ・ $a(0, 0)$ の書込みを行ない、
- ・ $a(i, j)$ の書込みを行なったならば、続いて $a((i+1) \lambda, (j+1) n)$ の書込みを行ない、
- ・ $a(\lambda-1, n-1)$ の書込みを行なったならば、終了する

という規則によって規定されるシーケンスに沿って、斜め方向にインターリーブすることにより、位相幾何学的に円環体を成すアレィを形成するようにした、ことを特徴とする方法。

【請求項9】 請求項8記載の方法において、該方法は更に、 n の値と λ の値とを、それら値が互いに素であるという条件を満足する任意の所望の整数値に選定するステップ、を含んでいることを特徴とする方法。

【請求項10】 ECC符号に従いインターリーブされた複数のコード化済シンボルから成る複数の符号語を復号化する復号器の能力を、冗長度を増大することなく強化する、復号器能力強化方法において、複数の符号語が矩形ブロックの中に並ぶように符号化を行ない、その際に各ブロックにおいて、個々の符号語がそのブロックの中の各々の横列を成すようにすると共に、夫々の横列を成す符号語の間で互いに対応するシンボルどうしが縦方向に位置が揃うようにする、符号化ステップと、

一度にブロック1つずつのインターリーブを行ない、その際に、そのブロック中の連続する複数の符号語の複数のシンボルを、1つの連続した斜めのシーケンスに沿ってインターリーブすることにより、位相幾何学的に円環体を成すアレィを形成し、それによって、全てのバースト誤りが、前記アレィ中のそれらバースト誤りの発生位置にかかわらず符号語から符号語へと連続するようにするインターリーブステップと、

全ての符号語を復号化し、全ての訂正可能な誤りを訂正し、前記ECC符号の誤り訂正能力を超える符号語のみをフラグで標識付けしてフラグ付符号語とする、復号化ステップと、

4

前記フラグ付符号語の各々について、当該フラグ付符号語に対して時間的に隣接した少なくとも1つの符号語を検査して、その隣接符号語中の誤り状態におけるロケーションであって、訂正がなされてその復号化が成功したロケーションを識別する、検査ステップと、

当該フラグ付符号語中のロケーションのうち、その隣接符号語中の誤り状態にあったロケーションに対して、斜め方向の連続シーケンスの中であって時間的に隣接した、特定ロケーションに存在する誤りを暫定的に消滅痕であると宣言する、宣言ステップと、

前記復号化ステップを再実行して、当該フラグ付符号語を訂正して復号化することを試みる、再実行ステップと、

を含んでいることを特徴とする復号器能力強化方法。

【請求項11】 請求項10記載の方法において、前記矩形ブロックの各々が、シンボル n 個分の長さのコード化済符号語が入行並んだものから成るようにしてあり、且つ、 n の値と λ の値とが互いに素であるという条件を満足する任意の所望の整数値に選定されていることを特徴とする方法。

【請求項12】 請求項10記載の方法において、前記復号化ステップは、各々のブロックの復号化を、そのブロックの次のブロックの復号化を開始する前に完了させるようにしたことを特徴とする方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は符号の誤り訂正に関し、より詳しくは、ランダム誤りと共にかなり多くのバースト誤りを発生する伝送チャネルに対応できるようにした、インターリーブ式の誤り訂正符号の符号語を復号化する復号器の、復号化能力を強化するための方法に関する。

【0002】

【従来の技術】 データ中に、バースト誤りとランダム誤りの2種類の誤りが混在する場合に、それら2種類の誤りを訂正できるようにするための最も一般的な方法は、誤り訂正符号(ECC)の符号語を、インターリーブした符号語にしておくという方法である。従来のこのインターリーブ式の方法では、まず、(ブロック1つ分の)一連の符号語の、各々の第0バイトだけを縦方向に並べて符号化して行き、それら符号語の全ての第0バイトの符号化が完了したならば、続いてそれら符号語の第1バイトだけを同じく縦方向に並べて符号化して行き、以下同様にして、一連の複数の矩形ブロックの各ブロックごとに、その1つのブロックの中に、ブロック1つ分の一連の符号語の全てのバイトを書き込むようにしていた。一方、復号化の際には、ある特定の1つのブロックの中の全ての符号語を、そのブロックの1本の横列ごとに復号化し、そして次々と連続して横列1本ずつ復号化して行くようにしている。このインターリーブを行なうこと

によって、1つのバースト誤りに含まれる m 個の誤りが、連続する m 個の符号語の間に空間的に分散されるようになり、即ち、それら m 個の誤りの全て、或いは殆ど全てが、1つの符号語の中に入ってしまうのを避けられるようになる。このインターリーブ式の方法は、それが良好に機能する状況であれば、個々の符号語の誤り訂正能力を超えてしまうという事態を招くことはない。

【0003】インターリーブ式の様々な方法を記載した文献としては、クラークら(Clark et al.)による「デジタル通信のための誤り訂正符号(Error-correction Coding for Digital Communications)」1981, Plenum Press発行 等がある。広く一般的に採用されているECCの1つに、リード-ソロモン符号(RS符号)と呼ばれているものがある。RS符号は、有限体(ガロア体)上のバイト訂正符号であり、その有限体はGF

(q)で表わされ、ここで q は素数のべきを表わしている。この q の値として一般的に用いられているのは「 2^8 」という値である。また、あるリード-ソロモン符号が、 $[n, k]$ RS符号というように書き表わされているならば、それは、符号語の長さを n バイト(ただし $n \leq q-1$)とし、そのうちの情報バイトの個数を k バイトとしたRS符号である。この場合、冗長バイトの個数(バイト数) r は、 $r = n - k$ となり、また、符号の最小距離 d は、 $d = (n - k) + 1$ となる。 $[n, k]$ RS符号を用いて「誤り」と「消滅痕(erasures)」とを訂正する場合に、 $2s + t \leq (n - k)$ を満足する s 個の誤り及び t 個の消滅痕を訂正することができる。「消滅痕」とは、誤りの一種であって、その発生位置が既知の(ただし、その正しい信号値は未知の)ものである。

尚、ここでは、単に「誤り」という場合には、それは、「消滅痕」ではない、通常の誤りのことを意味するものとする。1個の誤りは略々2個の消滅痕に相当するため、誤りのうちのあるものを消滅痕として取扱うことによって、冗長度を増大させることなく復号器の能力を強化することが可能となる。

【0004】米国特許第4559625号は、誤りを消滅痕として取扱うことによって符号器の能力を強化しようとする予測式の方法を記載した従来例の文献としては、出願人の知る限りで唯一の文献である。この米国特許の方法は、以下の特性を有している。

(1) 符号語の長さ(n)を、通常採用されている128バイトの長さにした場合、この方法では、 $127 \times (n - 1)$ 個の符号語(余りにも大きな個数である)をインターリーブしなければ、完全な螺旋対称性が得られない。そして完全な螺旋対称性が得られない場合には、このバースト予測式の方法では、更に複雑度が上昇してしまう。

(2) ある符号語が訂正可能な誤りを含んでいたならば、その符号語に隣接した符号語の中の、その訂正可能な誤りに隣接したロケーション(隣接ロケーション)に

フラグで標識付けをし、そして、隣接した符号語の中の誤りを、隣接ロケーションに位置する消滅痕であるものと、自動的に予測するようにしている。

(3) 一連の符号語を符号化するには、それら符号語が、1列ごとに斜めにずれたブロックの中に並ぶようにしている。即ち、1つの符号語を1本の横列とし、連続する符号語が縦方向に並ぶようにするが、その際に、連続する2つの符号語の間では、それら符号語の中の対応するバイトどうしが横方向にずれるようにしている。

【0005】図1には、米国特許第4559625号に記載されている螺旋インターリーブ式のデータ書込みシーケンスが、単純化して示されている。この図1に示した書込みシーケンスでは、符号化の際に、バイトを垂直方向に次々と書き込んで行くようにしており、その書き込んで行く順序をバイト番号で表わすと、図示の如く、 $0'$ 、 $7'$ 、 $10'$ 、 $1'$ 、 $4'$ 、 $11'$ 、 $2'$ 、 $5'$ 、 $8'$ 、 $3'$ 、 $6'$ 、 $9'$ 、 $0''$ 、 $7'$ 、 $10'$ 、 $1''$ 、 $4''$ 、 $11'$ という順序である。従ってこれらバイトは、符号化された状態では、図1のブロック $m-1$ 、ブロック m 、及びブロック $m+1$ のように、斜めにずれた形のブロックの中に並んでいる。ブロック m を構成している複数の符号語はその各々の長さが n バイトであり、またインターリーブ深さが i である。ただし、符号語「 $0' \sim 3'$ 」と、符号語「 $4' \sim 7'$ 」と、符号語「 $8' \sim 11'$ 」とは、上下に並んだものどうしが互いに横方向にずれており、全体として斜めにずれた形となっている。このブロック m の、バイト $0' \sim$ バイト $11'$ が復号化されるときには、復号化横列A、復号化横列B、復号化横列Cの順で、各々の横列ごとに次々と復号化されて行く。それら全てが復号化されれば、続いて、ブロック $m+1$ のバイト $0'' \sim$ バイト $11''$ (これらのバイトは、図にはそのうちの一部しか示していない)が同じように次々と復号化されて行き、以下、同様にしてその他のブロックも復号化されて行く。この復号化の処理は、もし誤りが全く検出されなければ、いかなる遅延も生じることなく進められて行く。

【0006】ここで、バイト $2'$ (即ち、符号語A(符号語「 $0' \sim 3'$ 」)の第3番目のバイト)に誤りが検出されたものとする。このバイト $2'$ に対して垂直方向に隣接する位置にある、符号語Bのバイト $5'$ には、符号語AにECCが適用されてその誤りの訂正が行なわれた後に、フラグによって自動的に、消滅痕であることを示す標識付けがなされる。続いてこの符号語Bに対してECCが適用される。この符号語Bの誤り訂正が成功したならば、そのバイト $5'$ が、実際に誤りを発生していたか否かを調べる。もしバイト $5'$ が実際には誤りを発生していなかったのであれば、そのフラグを、このバイト位置から除去する。一方、バイト $5'$ が実際に誤りを発生していたのであれば、符号語Cにフラグで標識付けすることによって、そのバイト $8'$ (符号語Cの先頭の

バイト)が消滅痕であることを自動的に予測し、且つそれを宣言する。以上と同じプロセスを、後続のブロック $m+1$ を復号化する際にも反復して実行する。要するにこの米国特許においては、ある1つの符号語の中に誤りが発生したならば、その符号語の中の誤りの位置に対して垂直方向に隣接した、次の符号語の中の対応する位置が、消滅痕であるものと自動的に予測するようにしているのである。

【0007】

【発明が解決しようとする課題】しかしながらこの従来例の方法では、実際には符号語が訂正可能であったにもかかわらず、自動的にフラグ標識付けがなされて消滅痕であると推定されてしまい、この方法を採用したばかりに、ECCの能力を超えてしまうという不都合が発生するおそれがあった。

【0008】更にこの米国特許の方法では、螺旋対称性を得るためには $i = n - 1$ としておく必要があり、即ち、インターリーブ深さ i が、1つの符号語の中に含まれるバイト数 n よりも丁度1つだけ少ない値としておく必要がある。これが必要とされるのは、この米国特許の式の方法を用いたときに、最後の横列(図1の例では横列C)だけを特別の処理方式で処理せずとも良いようにしたいからである。例えば、もし $i < (n - 1)$ であったならば、螺旋の規則性が壊れてしまうことになる。即ち、図1において、もし仮に、インターリーブする横列を、AとBとの2列だけとしたならば、A、B、Cの3列としていたときには螺旋対称性を壊すことなく10から1'へ回れたのに対して、7から1'へ回る際に、どうしても螺旋対称性が壊れてしまう。従って、以上に説明した従来例と比較して、更に復号化の能力を強化したECCの符号語を復号化するための復合器能力強化方法が求められており、本発明の目的は、このような強化方法を提供することである。

【0009】

【課題を解決するための手段】この目的を達成するために、本発明では、復号化能力を次のようにして強化している。まず、①ECCの能力を超えた符号語に対してのみフラグで標識付けをするようにしている。また、② n と λ とが互いに素であるということのみを条件として、符号語を任意のバイト長さ(n)及び任意の符号語深さ(λ)のものとすることができるようになっている。そして、③連続する複数の符号語のコード化済バイトを、斜め方向の単一の連続したシーケンスを成すようにインターリーブして、位相幾何学的に円環体を成すアレイを形成するようにし、それによって全てのバースト誤りが、そのアレイの中におけるそれらバースト誤りの発生位置の如何にかかわらず、1つの符号語から次の符号語へと連続するようにしている。以上の諸特性によって、 λ 本以下の横列にしか影響を及ぼさないバースト誤りに付随する問題が確実に回避されており、これが可能であるの

は、以上のようにすることによって「最後の横列」も、また「先頭の横列」も存在しなくなるからである。

【0010】

【実施例】これより、図2以下の図面を参照しつつ、本発明の好適実施例について説明して行く。図2は、本発明の実施例に係る装置10を示したものであり、この装置10はランダム誤りと共にかなり多くのバースト誤りを発生する通信チャネル上を伝送されるデジタル・データを、符号化及び復号化するための装置である。図示の如く、装置10は符号器12を含んでおり、この符号器12は、デジタル・データ入力を、符号語の構成要素であるシンボルのストリームへと変換するものである。符号語の1つ1つは、その長さを n バイトに定めてある。コード化済の符号語のシンボルに対しては、インターリーバー14によって、順序の並べ換えが行なわれ、この並べ換えの方式については後に重点的に詳述する。変調器16は、信号をチャネル18上で伝送できる形にするものであり、これを、インターリーブされた符号語をラン・レングス・リミテッド符号(例えば(1, 7)符号)へ変換することによって行なっている。

【0011】復調器20は、チャネル上を伝送されてきた信号を処理してデジタル・データの形にし、そこからシンボルのストリームを復元し(こうして復元されたシンボルのストリームは、チャネル上のノイズ等のために劣化している)、そして、その復元した、即ち復調したシンボルのストリームを、デインターリーバー22へ向けて出力する。デインターリーバー22は、インターリーブされている、そのシンボルのストリームに対して、インターリーバー14が行なった操作とは逆の操作を施すことによって、そのストリームの中のシンボルの並び順を、元の順序に戻す。こうして得られた符号語は、強化されたRS復号器24へ転送され、この復号器24は、従来のRS復号器では不可能な程に困難な、オリジナル・データの評価値の出力作業を、冗長度を増大させることなしに、ECC符号の誤り訂正能力を拡張することによって可能にしたものである。

【0012】尚、本明細書の発明の詳細な説明並びに請求の範囲において使用している「シンボル」という用語は、バイト、ハーフワード、ワード、或いはその他の任意の種類、所定個数のビットの集まりを、広く包括する用語である。ただし、理解を容易にするために、これまでの説明では、シンボルがバイトである場合について述べており、また、これ以降の説明でも、シンボルがバイトである場合について述べることにする。更に、以下の説明の中で使用する「 $\langle m \rangle_n$ 」という表記は、 n を法として m と合同な整数 i であって ($m \equiv i \pmod{n}$)、 $0 \leq i \leq n-1$ である整数 i を表わすものである。

【0013】本発明を実施する上での最初のステップは、 n と λ とが互いに素でなければならないということ

9

のみを条件として、長さが n バイトのコード化済の符号語を、インターリーブ深さ λ でインターリーブすることができるようなECC符号を選択することである。例えば、 $n=128$ 、 $\lambda=9$ とすれば、「128」と「9」とはいずれもそれ自体は素数でないが、 n と λ とは、互いに素である。従って、容易に理解されるように、 $\lambda=3, 5, 7, 11$ 、等々のように、 λ の値それ自体を素数にした上、 n をその λ では割り切れない数とすれば、上記の条件は簡単に満足することができる。

【0014】このECCは、複数の矩形ブロック（例えば、ブロック $m-1$ 、ブロック m 、ブロック $m+1$ ）の形に構成されるものである。それらブロックの単純化した具体例を示すならば、例えば図3に示したようなブロックであり、同図において各ブロックは複数の符号語から成り、それら符号語は、その各々の長さ n を4バイトとしてあり、またブロックの深さ λ を符号語3列分の深さとしてある。各々のブロックは、 $a(i, 0)$ 、 a

$(i, 1)$ 、...、 $a(i, n-1)$ で表わされる横列から成り、ここで、 $0 \leq i \leq \lambda-1$ である。本発明によれば、インターリーバ14は、1つ1つのブロックごとに、そのブロックの中の λ 本の横列を、 $(\lambda+n-1)$ 本の斜線へと並べ換え、即ち変換するものであり、これを次の規則に従って行なうようにしている。 a

$(0, 0)$ の書き込みを行なう。 $a(i, j)$ の書き込みを行なったならば、続いて $a(\langle i+1 \rangle \lambda, \langle j+1 \rangle n)$ の書き込みを行ない、ただし、 $0 \leq i \leq \lambda-1$ 、 $0 \leq j \leq n-1$

$a(\lambda-1, n-1)$ の書き込みを行なったならば、終了する。 n と λ とは互いに素であるため、本発明の方法によれば、ブロックの中の全てのバイトの書き込みを必ず完了することができる。

【0015】上記規則によって規定されたこの斜め方向インターリーブ法を実行することにより形成されるレイ・パターンは、位相幾何学的に円環体（ドーナツ形）を成すものであり、即ち、ドーナツに紐を斜めに巻付けて行き、その紐がドーナツを何周かした後に、その末端を先端に結び付けた形に対し、相似の形になる。位相幾何学的にこのような形となっているため、 λ 本以下の横列にしか影響を及ぼさないバースト誤りは連続になる。ここで連続という意味は、従来のインターリーブ法では最後の横列において不連続性が存在していたのに対し、このインターリーブ法では、最初の横列も最後の横列も存在しないということである。例えば、図3に示したブロック m 、ブロック $m+1$ 、... 中のシンボルは、1度にブロック1つずつ（即ち、1つのブロックの処理が完了してから次のブロックに移るようにして）インターリーブされることによって、それらの順序は、 $0'$ 、 $5'$ 、 $10'$ 、 $3'$ 、 $4'$ 、 $9'$ 、 $2'$ 、 $7'$ 、 $8'$ 、 $1'$ 、 $6'$ 、 $11'$ 、 $0''$ 、 $5''$ 、 $10''$ 、... というようになる。

10

【0016】こうしてインターリーブされた複数の符号語は、チャンネル18上を伝送された後に、デインターリーブされて、インターリーブされる前の順序に戻される。これは、それら符号語がコード化された直後の状態に戻されるということである。この後、それら符号語の復号化を行なう。この復号化はブロックごとに行ない、即ち、1つのブロックの復号化が完了したら次のブロックへ移るというようにして、しかも各ブロックの中では、各々の横列ごとに次々と連続して行なうようにしている。具体的には、例えば先ず、ブロック m について、その中のバイト $0'$ ～バイト $11'$ を各横列ごとに次々と連続して復号化して行き、それら全ての復号化が完了したならば、続いて、ブロック $m+1$ について、その中のバイト $0''$ ～バイト $11''$ を同様に復号化して行き、以下同様に処理して行く。この復号化の処理は、訂正不可能な誤りが検出されない限り、いかなる遅延も生じることなく進められて行く。

【0017】本発明によれば、後に更に詳細に説明するように、冗長性を更に増大させることなくRS復号器24の復号化能力を強化することができ、この復号化能力の強化は、以下にようにすることによって達成されている。

(1) 各ブロックの中の、以上に説明したインターリーブ処理を施したECC符号の全ての符号語（横列）を、次々と連続して復号化して行くようにする。

(2) ある1つのブロックにおいて、その中の全ての符号語の復号化が完了し、かつそれら符号語に関する、ECC符号の誤り訂正能力の及ぶ限りの誤り訂正とが完了したならば、そのブロックの中の、ECC符号の誤り訂正能力を超えてしまったために訂正できずに残っている符号語（即ち、誤りを含んだままの符号語）の全てに、フラグで標識付けをする。

(3) フラグで標識付けした符号語の各々について、その符号語に隣接している符号語（即ち、その符号語に先行する符号語ないし後続の符号語）のうちの、少なくとも一方の符号語（以下、「隣接符号語」という）を調べて、その「隣接符号語」の中に誤りが存在していたか否か（即ち、その「隣接符号語」が、復号化の際に誤りが発見されたためにその誤りの訂正が行なわれた複合語であるか否か）を判別する。もしそこに誤りが存在し、従ってその「隣接符号語」が、誤り訂正が行なわれたものであったならば、その誤り訂正が行なわれた「隣接符号語」の中の、誤りが生じていた全てのロケーションを特定する。

(4) そのフラグで標識付けした符号語の中のロケーションのうち、「隣接符号語」の中の誤りが生じていたロケーションに対して、斜め方向に隣接している（この「斜め方向に隣接している」というのは、上掲の規則の順番において連続しているという意味である）ロケーションに対して、そのロケーションが消滅痕であるとする

宣言を暫定的に行なう。これによって、そのフラグで標識付けした符号語の中の誤りの全てが（或いは、全てでなければそのうちの一部分が）消滅痕であると暫定的に宣言される。

（５）そのフラグで標識付けした符号語の復号化を試みる。

（６）その復号化に成功したならば、同じくフラグで標識付けておいたその他の符号語に対して、上記ステップ（３）、（４）、（５）を、反復して実行する。一方、復号化に成功しなかったならば、その符号語が訂正不可能な符号語であることを宣言し、復号化の再試行或いは終了等の、所定のプロトコルを起動する。

【００１８】本発明を実施するための、様々な具体的な方法については、以下に示す具体例と、そのフローチャートとから明らかにする。

具体例Ⅰ

この具体例Ⅰでは、使用する〔 n , k 〕 RS 符号は、その最小距離を $d=8$ とし、３つまでの誤りを訂正でき、４つまでの誤りを検出できるようにした RS 符号であるものとする。また、符号語の長さは $n=17$ であり、インターリーブ深さは $\lambda=6$ であるものとする。更にこの具体例Ⅰにおいては、図４に示したように、発生した誤りのパターンは、３つのバースト誤り b_1 、 b_2 、 b_3 と、３つのランダム誤り R とを含んでいるパターンであったものとする。復号器２４は、図５のフローチャートの中の諸々のステップに従い、横列の復号化と、その横列の中の誤り訂正とを行なっていくが、その際に、最初から復号化及び誤り訂正を行なえるのは、横列０、横列２、横列３、及び横列５だけであり、その理由は、これら横列だけは、その各々に含まれている誤りの個数が、上記のように選定した RS 符号の誤り訂正能力を超えていないからである。一方、残りの横列１と横列４とには、フラグで標識付けがなされ、なぜならば、これら２本の横列は、この最初の復号化の段階では、その誤りの訂正が不可能だからである。続いて復号器２４は、横列０のバイト２とバイト３とに誤りが発生していたため、暫定的にそれら誤りがバースト誤りであったものと推定し、そして横列１のバイト３とバイト４とが、消滅痕 b_1 と消滅痕 b_2 とであるものと宣言する。この宣言をした後の横列１の誤りの状態は、誤り訂正能力の範囲内にあるため、この段階で、復号器２４は、この横列１の２つの誤り b_3 及び R と、２つの消滅痕 b_1 及び b_2 とを訂正する。同様に、横列３の中の誤りが、バイト０、バイト１０、及びバイト１６であったことから、次の横列４の中のバイト１、バイト１１、及びバイト０は、夫々、消滅痕 b_1 、 b_2 、及び b_3 であるものと推定される。このように推定された後には、復号器２４は、横列４に関しては１つの誤り R と３つの消滅痕 b_1 、 b_2 、 b_3 とを訂正すれば良いことになり、これは可能であるため、それらを訂正する。このようにして、冗長性を更に

増大させることなく復号化能力を強化することができるという、望ましい結果が得られる。

【００１９】更に詳しく説明すると、この復号化の方法は、配線して組立てたハードウェアのロジックによって実現することもできるが、図５のフローチャートに示したように、ソフトウェアによっても実現することのできるものである。図５のフローチャートのステップ１１０では、復号器２４が、その入力として、デインターリーバ２２から λ 個の受信語 $R(0)$ 、 $R(1)$ 、...、 $R(\lambda-1)$ を受け取る。この入力ステップ１１１へ受け渡され、このステップ１１１では、フラグ集合 F を空集合にセットすると共に、１－カウンタを「０」にセットする。フラグ集合 F は、訂正不可能な受信語のインデックスを格納するための（即ち、訂正不可能な受信語にフラグで標識付けするための）集合であり、一方、１－カウンタは、処理中の受信語の番号を保持しておくためのカウンタである。このステップ１１１の処理で得られた出力は、ステップ１１２へ受け渡され、このステップ１１２では、現在受信語 $R(1)$ の復号化を試みる。この現在受信語 $R(1)$ において誤り訂正が可能であったならば、誤り集合 $E(1)$ をステップ１１３へ受け渡し、このステップ１１３では誤り集合 $E(1)$ の格納を行なう。一方、現在受信語 $R(1)$ において誤り訂正が不可能であったならば、ステップ１１６へ進み、フラグ集合 F にインデックス１を追加する。ステップ１１３で得られた出力と、ステップ１１６で得られた出力とは、ステップ１１４へ送られ、このステップ１１４では、インデックス１を「１」だけインクリメントする。このステップ１１４からは、ステップ１１５へ進み、このステップ１１５では、１の値が λ の値に達したか否かを調べる。これは、換言するならば、全ての受信語の処理が完了したか否かを調べていることに他ならない。もし、全ての受信語の処理が完了していなかったならば、ステップ１１２へ戻り、このステップ１１２からステップ１１５までの処理を反復して実行する。一方、最後の受信語の処理が完了していたならば、その処理結果をステップ１１７へ受け渡し、このステップ１１７では、フラグで標識付けされている受信語が存在しているか否かを調べる（即ち、フラグ集合 F を調べる）。もし、フラグで標識付けされている受信語が１つも存在していなかったならば、ステップ１１８において、訂正済符号語 $C(1)$ を、この復号器２４の出力として送出する。一方、フラグで標識付けされている受信語が存在していたならば、フラグ集合 F の内容を、ステップ１１９へ受け渡し、このステップ１１９から、強化復号化処理のプロセスを開始する。

【００２０】ステップ１１９では、受信語の全てにフラグで標識付けされているのか否かを調べる。もしそうであれば、強化復号器であっても処理は不可能であるため、ステップ１２０において、訂正不可能な誤りである

ことを宣言するための所定のプロトコルを起動する。一方、フラグで標識付けされていない受信語が存在していたならば、ステップ121において、フラグ集合Fの中の第1番目の要素jを判別する。続いてステップ122において、その判別した要素jに対応した受信語R

(j) にとっての、先行受信語R ($\langle j-1 \rangle$) の誤り集合 (即ち、誤り集合E ($\langle j-1 \rangle$)) が、空集合であるか否かを調べる。もしその誤り集合が空集合であったならば、誤り集合E (j) の状態改善を行なうことは不可能であるため、ステップ120へ進んで、訂正不可能な誤りであることを宣言するための所定のプロトコルを起動する。一方、誤り集合E ($\langle j-1 \rangle$) が空集合ではなかったならば、ステップ123において、この誤り集合E ($\langle j-1 \rangle$) に基づいて、受信語R (j) の中の $\langle 1 + E (\langle j-1 \rangle) \rangle_n$ で表わされる夫々のロケーションに誤りがあるものと予測する。それらロケーションは、この受信語R (j) の中にあるロケーションのうち、集合E ($\langle j-1 \rangle$) に含まれている位置に対して斜めに隣接した位置に相当するロケーションである。復号器24は、それら予測ロケーションが消滅痕であることを宣言した上で、ステップ124において、受信語R (j) の復号化を試みる。もし受信語R (j) が、それらロケーションが消滅痕であると宣言しても尚、復号化の不可能なものであったならば、ステップ120へ進んで上述のプロトコルを起動する。一方、受信語R (j) の復号化が、その消滅痕の宣言によって可能になったならば、ステップ125において、この受信語R (j) の誤り集合E (j) を格納し、続いてステップ126において、フラグ集合Fから要素jを抜き取った後に、ステップ117へ戻り、そしてこのステップ117から、以上の強化復号化処理のプロセスを反復して実行する。

【0021】具体例II

容易に理解されるように、以上の具体例Iのプロセスでは、フラグで標識付けされているある横列の中に多数のランダム誤りRが含まれていた場合には、その横列に基づいて、後続の横列において消滅痕を宣言したならば、宣言された消滅痕のうちに、多くの偽消滅痕が含まれてしまうことになる。そして、このような事態が生じると、ECC符号の誤り訂正能力を超えてしまったり、誤った訂正が行なわれてしまうおそれがある。

【0022】この具体例IIでは、具体例Iで使用しているものと同じ $[n, k]$ RS符号を使用するものとするが、ただしここでは、発生した誤りのパターンは、図4の誤りパターンとはやや異なった、図6に示したパターンであったものとする。具体例Iの場合と同様に、ここでも復号器24は、4つまでの誤りを検出することができ、3つまでの誤りを訂正することができる。復号器24は、最初の復号化の際に、横列0、横列1、横列3、横列4、及び横列5に関しては誤り訂正を行ない、横列

2には、フラグで標識付けをする。続いて符号器24は、横列1のバイト4とバイト8とに誤りが発生していたことから、横列2のバイト5とバイト9とが消滅痕であると宣言する。この結果、この横列2には、3つの誤り (バイト2、12の2つのR、それにバイト15のb1) と、バイト5の (偽の) 消滅痕b2と、バイト9の消滅痕b3とを含むことになる。誤りと消滅痕との組み合わせがこのようになっている場合には、図5のフローチャートに関して説明したステップでは、その誤りと消滅痕との組み合わせを、検出することはできるが、訂正することはできない。何故ならば、このような組み合わせは、このECCの誤り訂正能力を超えているからである。

【0023】そこで、この問題を克服し、誤った訂正が行なわれる確率を大幅に低減することのできる、具体例Iに示したものよりも複雑な2通りの復号化方法の実施例について、以下に説明する。尚、図7と図8とに示した夫々の復号化方法において、図5の復号化方法の中のステップと同一内容のステップには同一の引用符号を付してあり、また、図5の復号化方法の中のステップとはやや内容が異なるステップには、図5で使用した引用符号にダッシュ (図7) ないしダブル・ダッシュ (図8) を加えた引用符号を付してある。

【0024】図7に示した復号化方法は、そのステップ123'の手前までの部分は、図5に関して説明した復号化方法と同じであるが、このステップ123'では、図5のステップ123よりも更に複雑な処理を実行するようにしている。即ち、このステップ123'では、誤り集合E ($\langle j-1 \rangle$) の部分集合Aの各々について、受信語R (j) の中の $\langle A+1 \rangle_n$ で表わされるロケーションを消滅痕であると宣言した上で、受信語R (j) の復号化を試みるようにしている。そして、ある部分集合Aについて、受信語R (j) の復号化が可能であったならば、その部分集合Aに対応した誤り集合E (A) を格納する。そして、こうして得られた (通常は複数の) 誤り集合E (A) の内容をステップ124'へ受け渡し、このステップ124'では、それら複数の集合E (A) のうちに多数派が存在しているか否か (即ち、それら集合E (A) のうちに、互いに同一の多数の集合が存在しているか否か) を判断する。もし集合E (A) のうちに多数派が存在していなかったならば、ステップ120へ進んで、訂正不可能な誤りに対処するためのプロトコルを起動する。一方、集合E (A) のうちに多数派が存在していたならば、ステップ125'において、その多数派を誤り集合E (j) として該誤り集合E (j) を格納し、これ以後、図5の復号化方法と同様に処理を続行する。

【0025】図8に示した復号化方法は、そのステップ122'及びステップ123''以外の部分は、図5に関して説明した復号化方法と同じである。ステップ12

2"では、フラグで標識付けされている受信語のうちの最初の受信語 $R(j)$ に対して先行する受信語 $R(\langle j-1 \rangle_1)$ と後続の受信語 $R(\langle j+1 \rangle_1)$ の、夫々の誤り集合 $E(\langle j-1 \rangle_1)$ と $E(\langle j+1 \rangle_1)$ とを共に調べる。もしそれら誤り集合のうちのいずれか一方が空集合であったならば、ステップ120へ進んで、訂正不可能な誤りに対処するためのプロトコルを起動する。一方、それら誤り集合がいずれも空集合ではなかったならば、ステップ123"へ進む。このステップ123"では、受信語 $R(j)$ の中のロケーションのうち、先行する符号語 $R(\langle j-1 \rangle_1)$ の誤り集合 $E(\langle j-1 \rangle_1)$ の中のいずれかの要素の位置に対しても斜めに隣接し、また、後続の受信語 $R(\langle j+1 \rangle_1)$ の誤り集合 $E(\langle j+1 \rangle_1)$ の中のいずれかの要素の位置に対しても斜めに隣接した位置（この「斜めに隣接した位置」とは、前述のインターリーブの規則に従って書込みをおこなうときの連続した斜めの位置のことである）に相当するロケーションを、消滅痕であると宣言する。換言すれば、受信語 $R(j)$ のロケーションのうちで消滅痕であると宣言されるロケーションは、 $\langle E(\langle j-1 \rangle_1) + 1 \rangle_n$ で表わされるロケーション集合と、 $\langle E(\langle j+1 \rangle_1) - 1 \rangle_n$ で表わされるロケーション集合との、交わり集合 (\cap) に含まれるロケーションである。これ以後、この図8の復号化方法は、図5の復号化方法と同様に処理を続行する。

【0026】

【発明の効果】本発明は、以上説明したような方法でデ

ータを符号化し、かつインターリーブして伝送し、そして受信側においてデインターリーブし、復号化するように構成されているので、従来例のものと比較して、ECCの符号語を復号化する能力を強化することができ、しかも螺旋対称性が壊れることがないので複雑な処理を必要としない、という作用効果を奏することができる。

【図面の簡単な説明】

【図1】従来例である米国特許第4559625号に記載されている螺旋インターリーブ法のデータ書込みシーケンスを、単純化して示した説明図である。

【図2】通信チャネルを介して伝送されるデジタル・データを符号化及び復号化するための、本発明の実施例に係る装置のブロック図である。

【図3】本発明の実施例に係る円環体インターリーブ法のデータ書込みシーケンスを、単純化して示した説明図である。

【図4】一具体例の誤りパターンの模式図である。

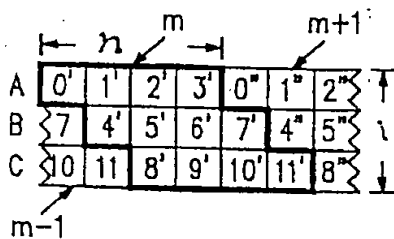
【図5】図4の誤りパターンを復号化することのできる、本発明の方法を示したフローチャートである。

【図6】図4の誤りパターンとは幾分異なった、別具体例の誤りパターンの模式図である。

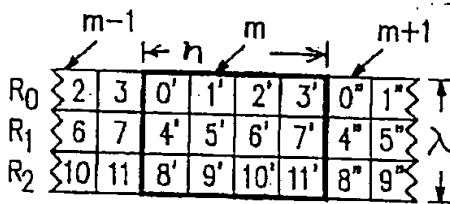
【図7】図6の誤りパターンを復号化することのできる、本発明の方法を示したフローチャートである。

【図8】図6の誤りパターンを図7の方法とは別の方法で復号化することのできる、本発明の方法を示したフローチャートである。

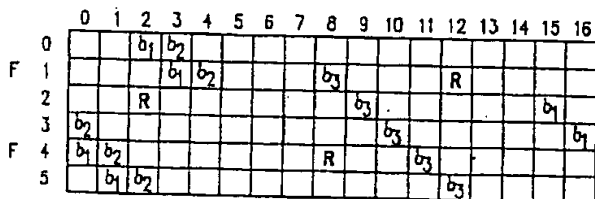
【図1】



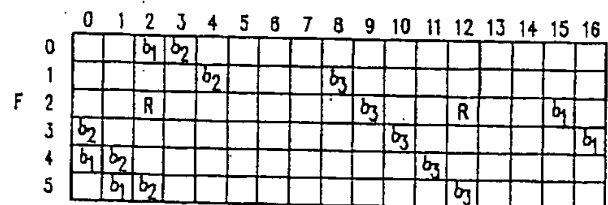
【図3】



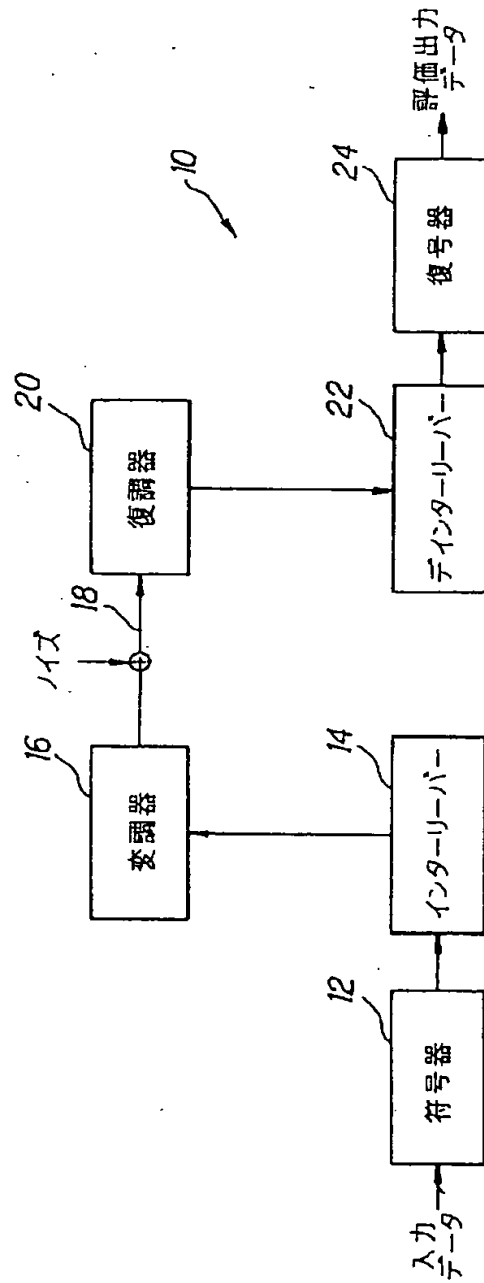
【図4】



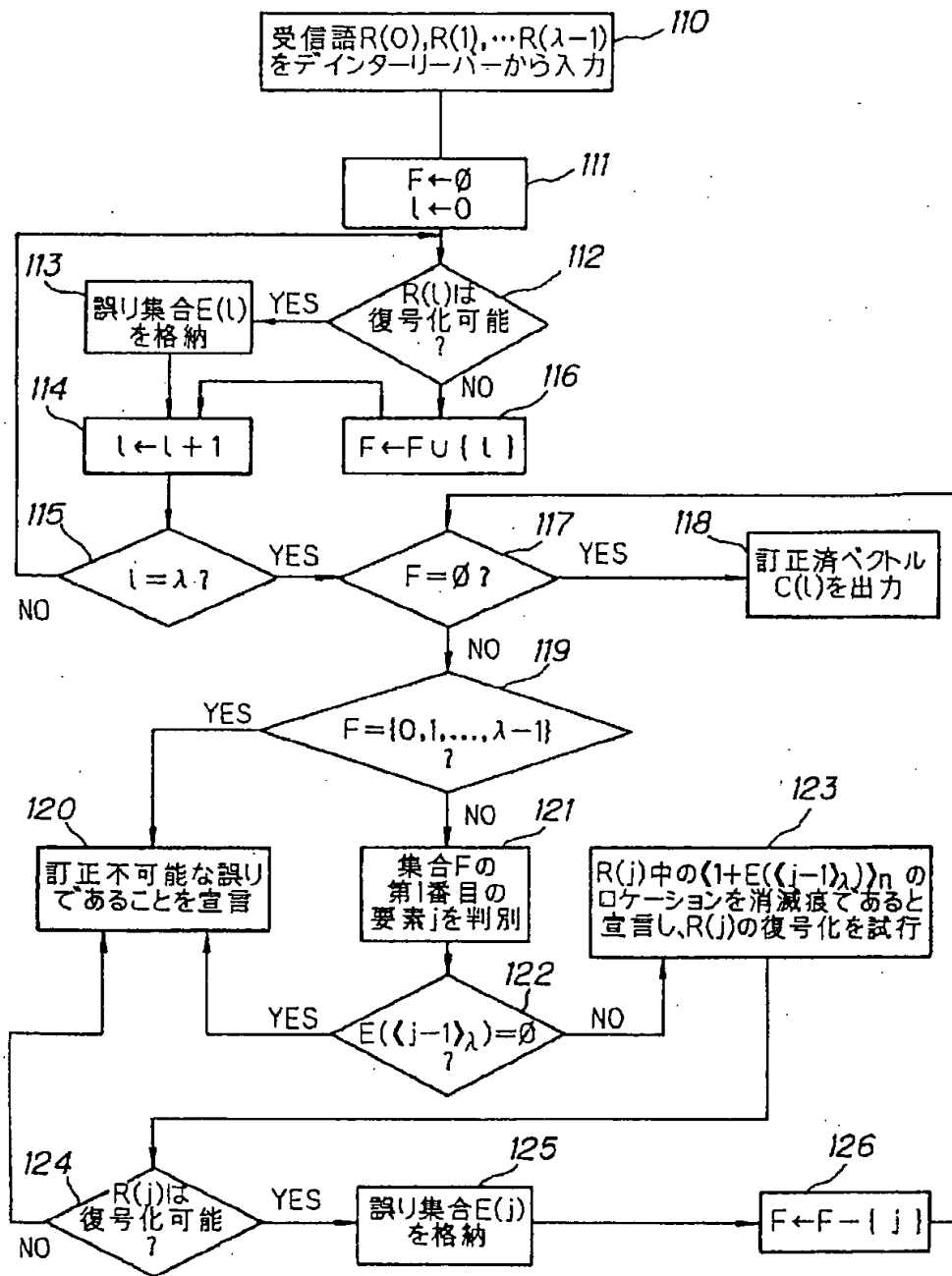
【図6】



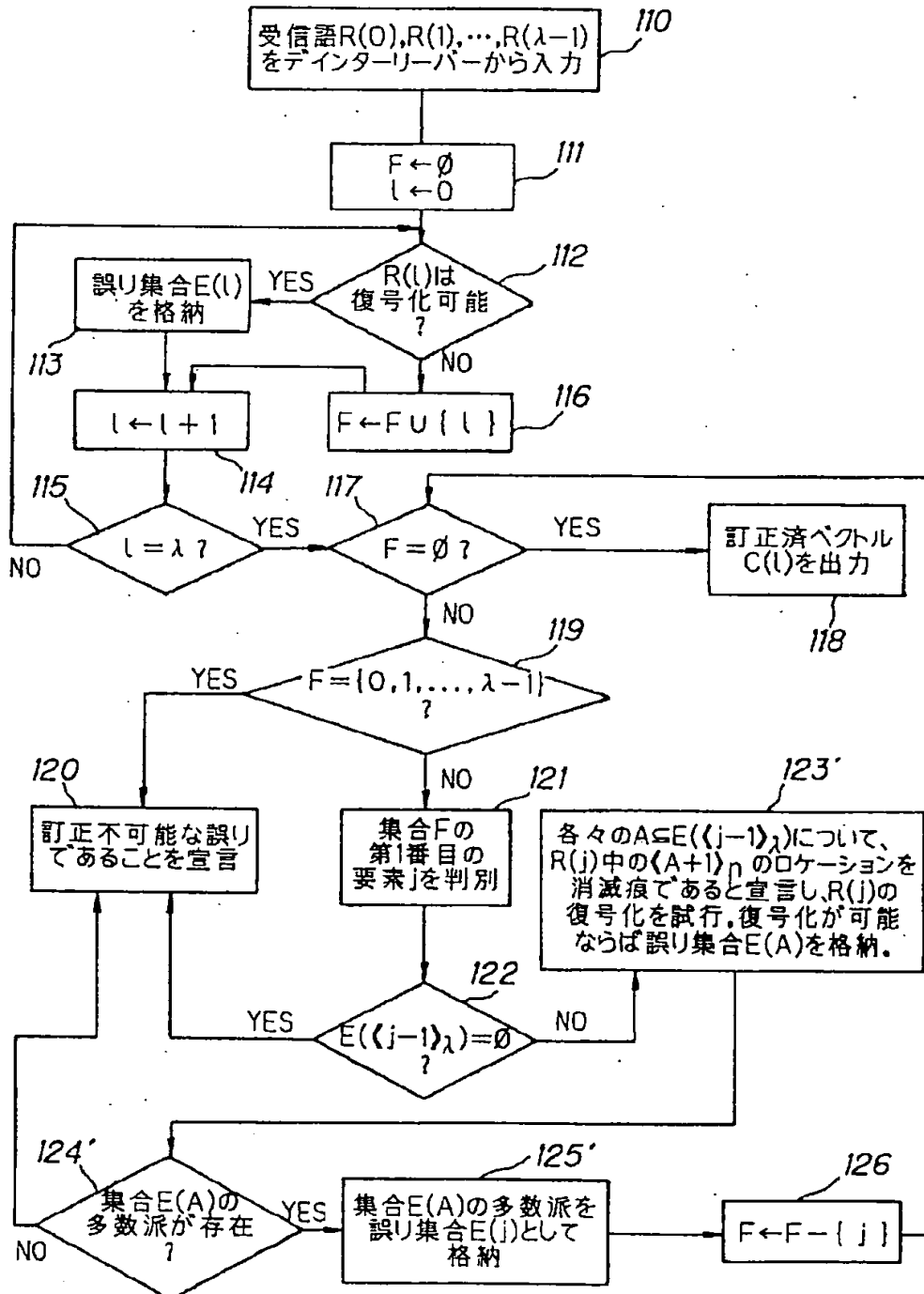
【図2】



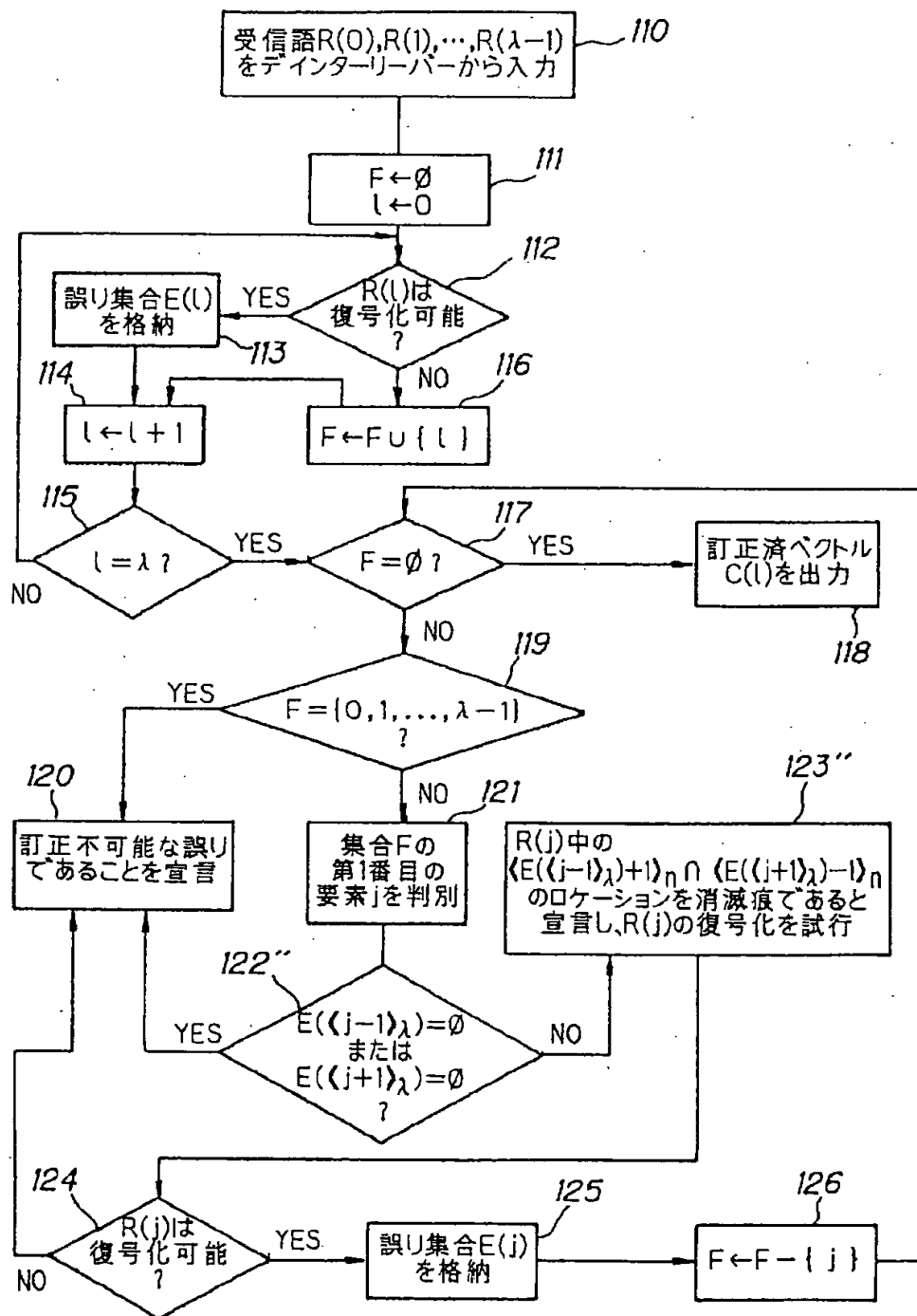
【図5】



【図7】



【図8】



フロントページの続き

(72)発明者 ヘンリクス・セー・ヴァン・ティルボルグ
オランダ王国5683 エル・イエー・ベス
ト、クラボトス 19番地